ACORN: A system for CVS macro design by tree placement and tree customization

by Peter S. Hauge
Ellen J. Yoffa

ACORN is a system for the physical design of cascode voltage switch (CVS) macros which utilizes tree placement and tree customization to improve macro wirability. The results obtained by designing a 43-tree differential (DCVS) macro on a masterslice chip image are presented to illustrate the design improvements. In this example, tree placement reduces wire length and via count by 12 percent relative to a transistor-pair placement design. Tree customization increases this improvement to 25 percent, and increases the porosity of the wired macro to vertical global wires from 8 percent for tree placement alone to 15 percent.

Introduction

This paper describes ACORN, a system for the design of cascode voltage switch (CVS) macros, which, by tree placement and tree customization, significantly increases the macro wirability. CVS circuits implement logic with fewer transistors than traditional NAND/NOR gates, and have shown in many cases leverage in performance as well as density [1].

The basic design blocks of CVS circuits are differential transistor pairs (differential, DCVS), or individual transistors (single-ended, SCVS). Other possible block types include clocks, load devices, and buffers. These components are connected to build trees, a collection of which in turn constitute a macro.

For electrical, timing, and physical design reasons, the logical tree height is limited to a specified maximum, typically of the order of eight or less. This represents the maximum number of transistors through which a signal passes in traversing a tree. Significantly for physical design considerations, this limitation permits the specification on a chip image of tree accounting areas, contiguous areas of constant height equal to that of the tallest tree, which are legal tree locations. This regularity permits local customization of the trees which enhances their wirability. This paper deals with DCVS macros, but ACORN accommodates SCVS designs as well.

There does not exist a one-to-one correspondence between a Boolean expression and a tree structure: in general, a given expression may be implemented by any of several structures. An important part of the ACORN design process is the selection of a tree from a set of possible implementations which can be widely diverse. By making this choice intelligently it is possible to reduce wiring congestion.

©Copyright 1984 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

The large number of input variables associated with each CVS tree, as well as the considerable fan-out of each variable, contributes to the complexity of inter-tree wiring. In order to alleviate this problem, chip images and device designs must provide easy access for these input variable wires and must also encourage straight-line wiring (busing) of common variables in adjacent trees. A masterslice chip image which has these characteristics is shown schematically in **Figure 1**. Each horizontal row of adjacent tree accounting areas forms a rail of trees in which the gate contacts to the transistors on each level are aligned. ACORN exploits this physical alignment of gates by customizing the trees: that is, by selecting implementations which produce the maximum possible number of gate variable matches on common physical levels across a rail. When wires must take even small jogs in order to make connections, large penalties result. The increase in horizontal variable busing obtained from customization leads to a significant decrease in wire length and via count, and an increase in the porosity to vertically oriented global wires.

ACORN provides the capability for wiring the primary inputs and outputs of the macro. ACORN also evaluates a completed macro design on the basis of its porosity to global wires. This measurement is done by assigning pairs of pins to opposite ends of each wiring track and attempting to wire either horizontally or vertically across the macro. This process determines the number of global wires which may be routed through the wired macro.

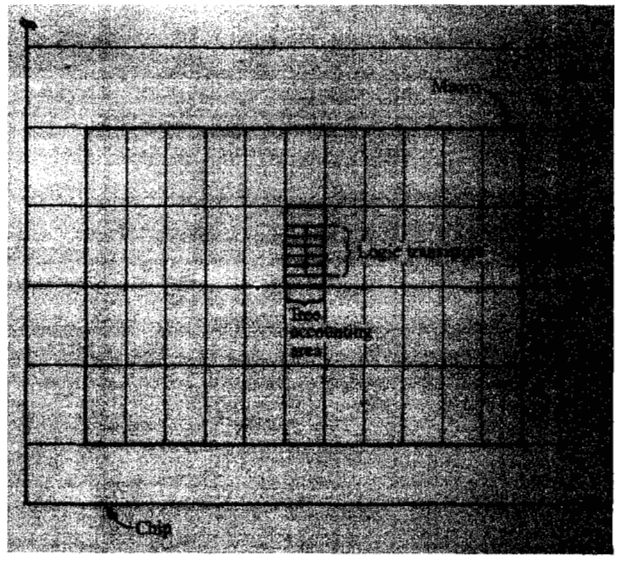

Figure 1

A masterslice chip image on which trees may be placed in contiguous tree accounting areas and subsequently customized to improve wirability. A macro area spanning 44 tree accounting areas (four rails high by 44 transistors wide) is shown on the image.

Concept of tree customization

Tree customization depends on the fact that the tree which instantiates a particular Boolean expression can assume any of several different configurations, differing in transistor count and/or variable ordering. Frequently a given variable may be moved to a desired level, without increasing the transistor count, by selecting an alternate tree configuration. In practice, each tree is generated independently during gate-level logic design, at which time there is no information about its local environment on the chip. Consequently, a choice of tree configuration is made with knowledge of only general criteria for selection: few transistors and simplicity of internal wiring. Other important criteria are environment-specific, such as the most advantageous level in the tree for a particular variable to be assigned. A procedure which chooses a structure for a tree before its environment is known does not provide an optimal design. After the trees have been placed, ACORN selects for each tree a configuration which has the internal ordering of transistor variables that best aligns variables for busing between adjacent trees, and thus optimizes the macro wirability.

As a simple illustration of the concept of tree customization, consider the two circuits shown in **Figure 2(a).** The blocks labeled *A* and *B* represent transistors having those variables respectively as gate inputs. Source and drain

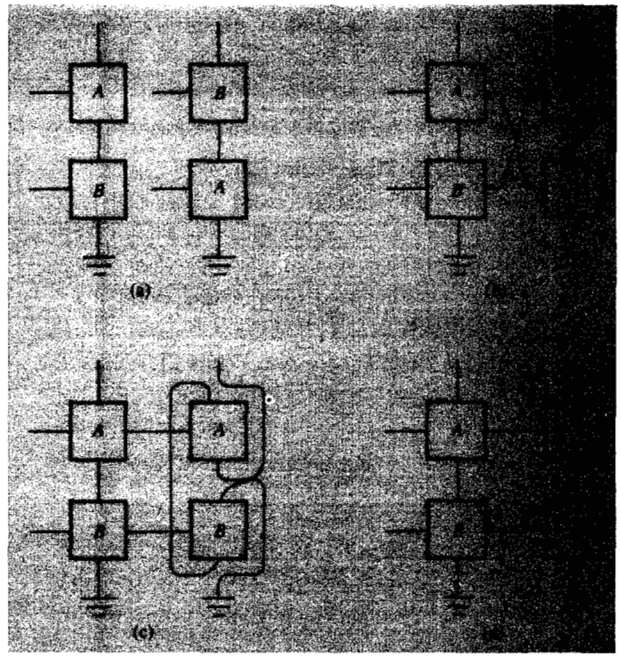

Figure 2

A pair of circuits which illustrate customization to improve inter-circuit (inter-tree) wiring; (a) As made. (b) Wired as made. (c) Transistors rearranged. (d) Customized.

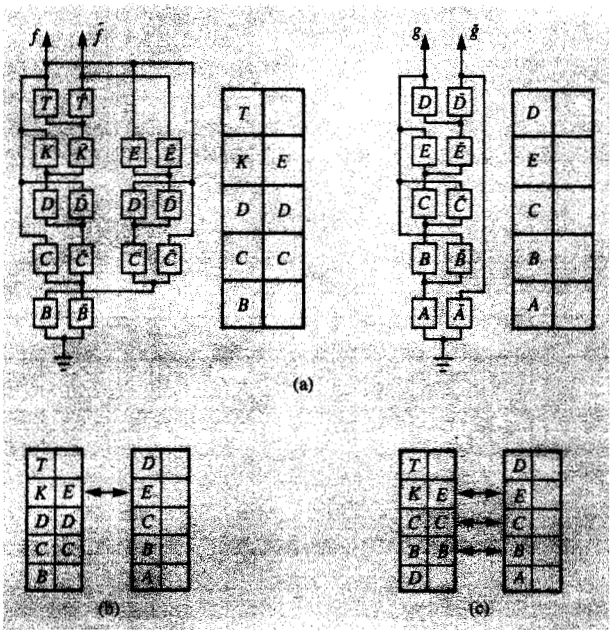

Figure 3

(a) Trees which implement the Boolean expressions $f = BC\bar{D}\bar{E} + \bar{B}\bar{C}DK\bar{T}$ and $g = \bar{A} + ABC\bar{E}\bar{D}$, shown as interconnected transistor pairs (left) and in a symbolic form (right) which indicates the placement of the pairs on a masterslice grid. (b) Contiguous placement of these trees, showing one level match (variable E). (c) Contiguous placement after customization of the first tree, showing matches on three levels (variables E, C, and B).

connections are at the bottom and top of the blocks, respectively. If these circuits are placed next to each other, the variables A and B are wired as shown in **Figure 2(b).** Although the internal wiring is simple, the wiring between the two circuits is complicated. On the other hand, if a placement program is free to place the transistors, it might choose to reorder them, as shown in **Figure 2(c).** Then the wiring between the two circuits is simplified, but this has been achieved at the expense of complicating the internal wiring of the second circuit. However, suppose that in the expression implemented by the second circuit the variables A and B are symmetric. The transistor gate inputs can then be interchanged, as shown in **Figure 2(d),** without changing the behavior of that circuit. The internal connectivity of the second circuit has been changed (e.g., the transistor with input B now connects to ground), but now both the internal and inter-circuit wiring is simplified.

Figure 3(a) shows two DCVS trees, composed of interconnected transistor pairs, and the Boolean expressions they implement: $f = BC\bar{D}\bar{E} + \bar{B}\bar{C}DK\bar{T}$ and $g = \bar{A} + ABC\bar{E}\bar{D}$. Next to each tree are the corresponding symbolic representations of the differential pair level assignments in five-differential-pair-high by two-wide tree accounting areas. Note that when these trees are placed adjacently, only one variable, E, occurs on the same level in both trees. The inter-tree wiring for that variable is simple, while that for the other variables is likely to be complex. However, in the first tree, the sets of variables $\{B, C, \bar{D}\}$ and $\{K, T\}$ are symmetric. Variable interchanges may be made within these sets without changing the function implemented by the tree. By interchanging symmetric variables B, C, and $\bar{D}$ as shown in **Figure 3(c),** two additional variable level matches are created, and the inter-tree wiring is correspondingly simplified.

Note that the process of symmetric-variable interchange is more powerful than that of pin swapping. This is clear from the above example, in which the number of connections in the signal nets has changed: For example, in **Figure 3(b),** the B net has two nodes, whereas in Fig. 3(c) it has three. The capability of some wiring programs to recognize interchangeable pins is not sufficient to achieve the full benefits of tree customization. Logically equivalent tree topologies occur which cannot be obtained from one another by a simple pin label interchange.

ACORN generates a set of allowed configurations for each tree from a given initial instantiation. This is done by interchanging symmetric variables as described above, by switching within a particular level the column locations of transistor pairs and by sliding transistor pairs to unoccupied levels within the tree. The second operation is important because variables in neighboring trees are aligned for busing *only* when the corresponding differential pairs are immediately adjacent, with no intervening pairs on that level.

To illustrate, we enumerate here the allowed configurations for the first tree in Fig. 3(a). As stated above, the Boolean expression f has two sets of symmetric variables $\{B, C, \bar{D}\}$ and $\{K, T\}$. Also, the assignment of variables K and E can be made in either order on level 4. Since all levels are occupied, no sliding can be done. Considering all possible symmetric variable interchanges ($6 \times 2 = 12$) and column switches (2), there are 24 allowed configurations for this tree.

Tree placement

In ACORN, entire trees are placed as single objects. The connectivity list is derived from the block-level connectivity by deleting all internal tree nets. The inter-tree connectivity indicates whether and how many times trees are connected to each other and to the primary inputs and outputs of the macro.

Placement of trees can be accomplished by any of a number of means [2–5]. ACORN uses a placement program which places the trees on the basis of inter-tree connections so as to minimize the total net length and to equalize the wiring congestion over the macro. Trees are thus placed

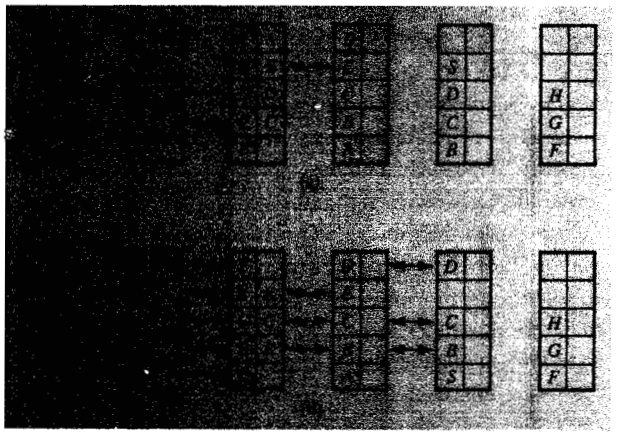

Figure 4

Rail optimization for a set of six contiguously placed trees: (a) Uncustomized trees. The configuration of each tree is that given by the tree generation program. There are two variable matches across the rail. (b) Customized trees. The configuration of each tree is selected by a rail optimization algorithm from a set of possible instantiations for each tree. The flexibility of each tree is governed by the properties listed in Table 1. The maximum number (16) of level matches across the rail is obtained.

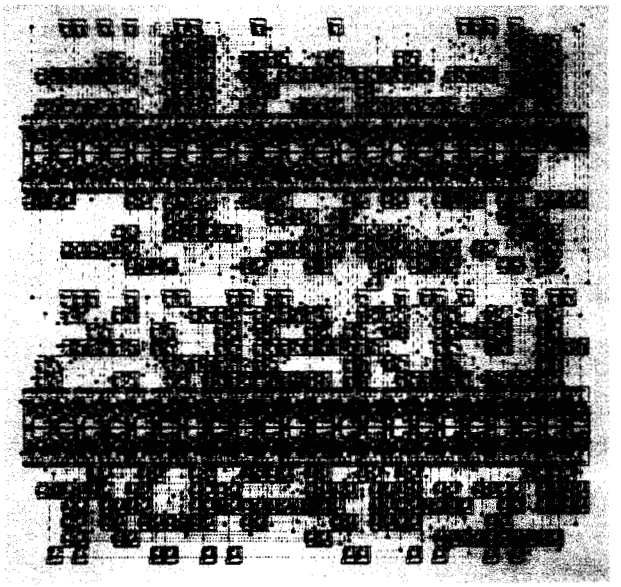

Figure 5

Transistor-level design of a 43-tree random logic DCVS macro. Macro size is four rails high (alternate rails are inverted) by 44 transistors wide.

without specification of their internal structure. In order to encourage variable busing, horizontal and vertical net lengths are weighted during placement in such a way that trees which share variables are preferentially placed in a common rail.

Rail optimization

To customize a rail of trees within the macro, ACORN invokes a rail optimization procedure which incorporates an algorithm guaranteed to maximize the amount of nearest-neighbor variable busing across the rail [6]. This algorithm uses as input the sets of possible configurations for each tree.

Figure 4(a) shows a rail of six trees in their initial (arbitrary) state determined by the tree generation program. (The trees of Fig. 3 occupy positions 3 and 4 on the rail.) **Table 1** shows the allowed customizing operations for each of these trees, and the resulting total number of possible configurations. Note that the only aligned variables in Fig. 4(a) are variables C on level 2 (trees 2 and 3) and E on level 4 (trees 3 and 4).

Figure 4(b) shows the rail of Fig. 4(a) after ACORN has customized the rail of trees. The maximum number (16) of level matches across the rail has been obtained.

Table 1 Allowed operations for customizing the six trees shown in Fig. 4. The Boolean expressions implemented, the sets of symmetric variables, possible column switches and row slides, and the total number of instantiations are indicated for each tree.

Tree	*Boolean expression*	*Symmetric variables*	*Column switches*	*Row slides*	*Number of configurations*
1	$B\bar{D}CU + B\bar{D}\bar{C}EG$	$\{E, \bar{G}\}, \{B, \bar{D}\}$ (4)	$\{E, U\}$ (2)	—	8
2	$BC\bar{K}T + \bar{B}\bar{C}DEG$	$\{B, C\}, \{E, G\}, \{K, \bar{T}\}$ (8)	$\{T, G\}, \{K, E\}$ (4)	—	32
3	$BCD\bar{E} + \bar{B}\bar{C}D\bar{K}\bar{T}$	$\{BC\bar{D}\}, \{KT\}$ (12)	$\{K, E\}$ (2)	—	24
4	$\bar{A} + ABC\bar{E}\bar{D}$	$\{\bar{B}, \bar{C}, D, E\}$ (24)	—	—	24
5	$\bar{B}\bar{C}D\bar{S}$	$\{B, C, \bar{D}, S\}$ (24)	—	(5)	120
6	$\bar{F} + G\bar{H}$	$\{G, \bar{H}\}$ (2)	—	(10)	20

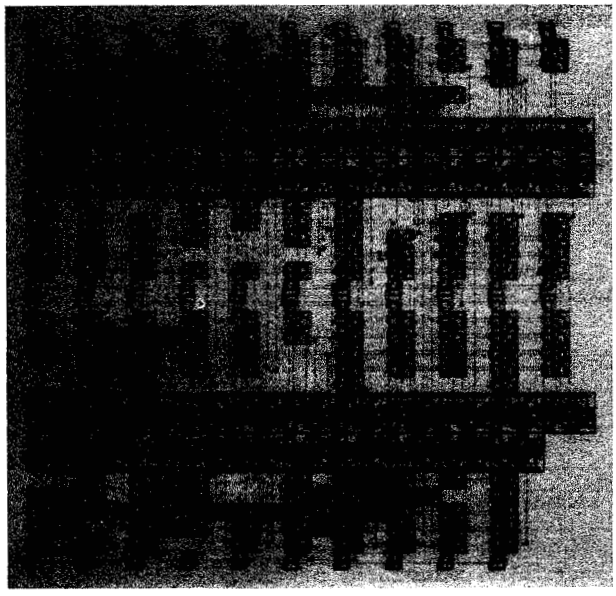

Figure 6

The 43-tree DCVS macro of Fig. 5 as designed by ACORN, in the same area, using tree placement and tree customization. Wire length and via count were reduced from the design of Fig. 5 by 25 percent.

Results of ACORN design

A straightforward method of CVS macro design is to use the block connectivity list to place and wire the individual transistor pairs. A random logic macro composed of 43 differential trees (224 differential pairs) designed in this manner on the image of Fig. 1 is shown in **Figure 5.** The blocks were placed in a rectangular region, four rails high and 44 transistors (11 tree accounting areas) wide. There are 477 nets (944 connections).

In transistor-level placement, all of the differential pairs in the macro are placed simultaneously with the objective of reducing the total net length and achieving uniformity of wiring congestion across the macro. The quality of this placement suffers from the inability of the placement program to evaluate candidate placements on the basis of local criteria. Moreover, the time required to place objects increases with the number to be placed. For chips and large macros, individual block placement is not feasible. In addition, transistor placement often permits members of a tree to be dispersed over a large area on the macro, with concomitant lengthening of intra-tree connections. Such results are unacceptable for electrical reasons. These objections to the transistor-level design method are avoided in ACORN by placing entire trees as single objects.

To demonstrate the improvements obtained by using ACORN tree placement and tree customization, we designed the macro of Fig. 5 in three other ways: (a) by placing the trees without customization, (b) by placing the trees and performing a limited customization, and (c) by placing the trees and performing a full customization. These cases are described below, and the improvements obtained are summarized in **Table 2.**

The tree generation program chooses an initial configuration for each tree which is an arbitrary selection from the possible choices. In case (a), these trees were placed, but not customized. The same tree placement was used for cases (b) and (c). The placement of uncustomized trees had a total of 15 variable level matches. Tree placement without customization decreased by 12 percent the wire length and via count from that of the transistor-level design.

Of the three customizing operations described above, only symmetric variable interchange changes the block connectivity; column switches and row slides do not. During macro physical design, it may be desirable to leave the connectivity unchanged. If so, ACORN can perform a limited customization, using only the last two actions. With tree placement followed by this limited customization, case

Table 2 Summary of the improvements obtained from ACORN designs over transistor level design for the 43-tree DCVS macro shown in Figs. 5–7. Design (a): Tree placement alone. Design (b): Tree placement plus limited (connectivity-preserving) customization. Design (c): Tree placement plus full customization.

Design	*Variable matches*	*Wire length (normalized)*	*Via count (normalized)*	*Vertical porosity (%)*
Transistor level	—	100	100	—
a	15	87	88	8
b	30	85	85	13
c	74	76	74	15

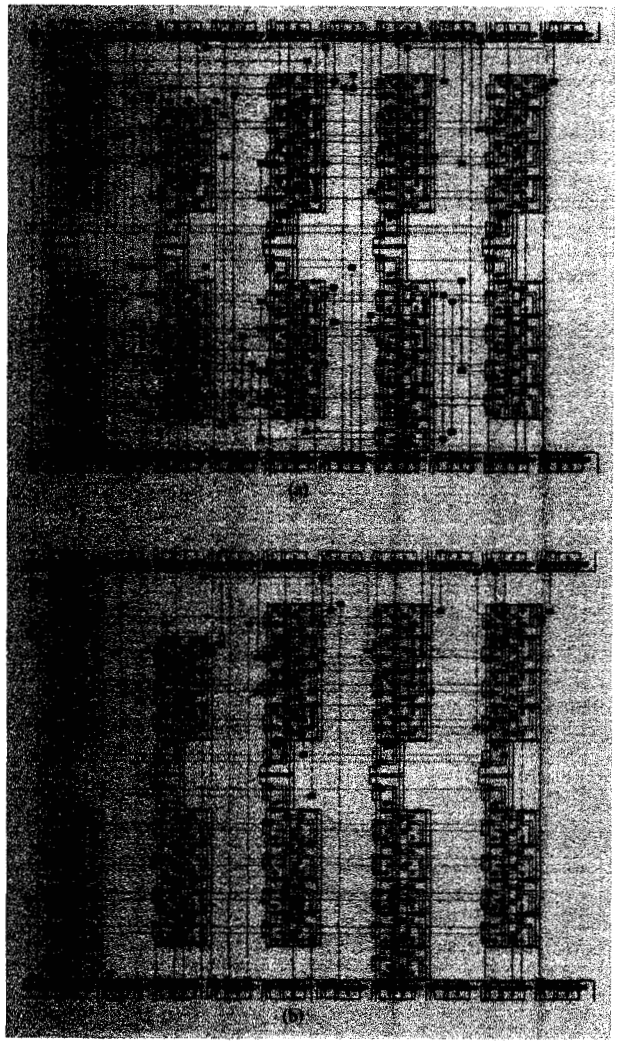

Figure 7

Enlargement of the right-hand portion of the second and third rails of ACORN designs of the 43-tree DCVS macro: (a) Trees were placed but not customized (full design is not shown.) (b) Trees were placed and fully customized (full design is shown in Fig. 6.) A significant decrease in wiring congestion and number of vias due to tree customization is seen in (b).

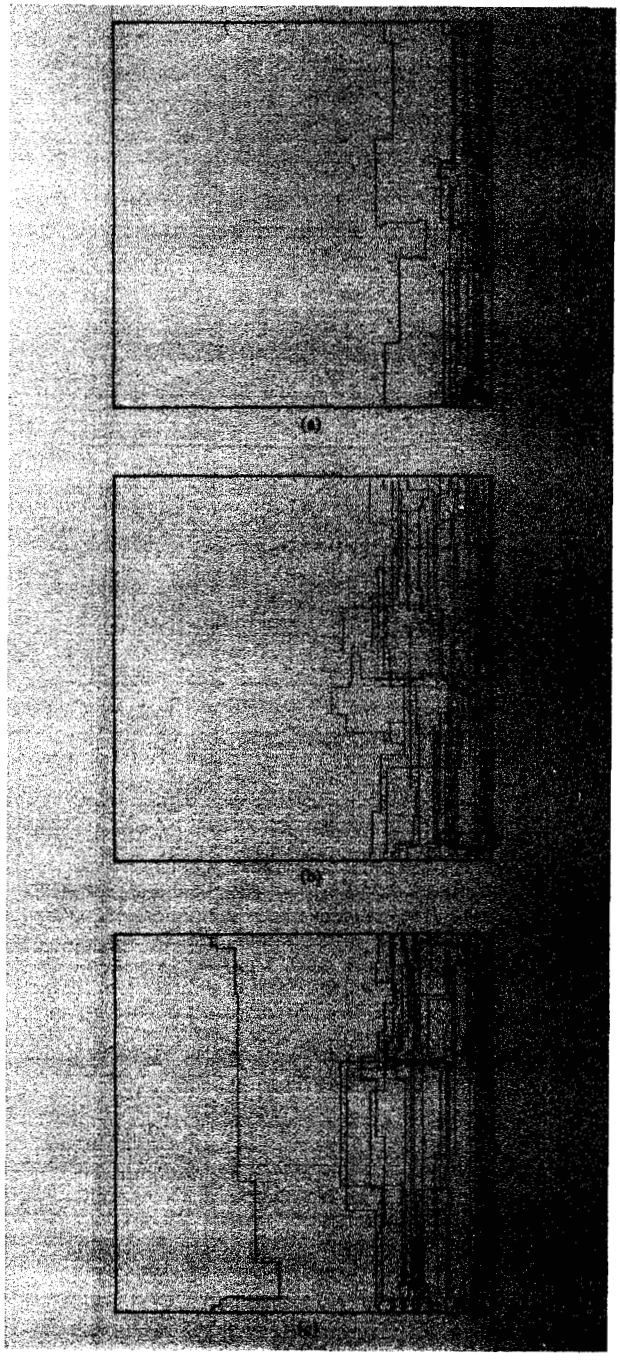

Figure 8

Vertically oriented global wires routed through the fully wired ACORN designs of the 43-tree DCVS macro: (a) Tree placement without customization. See Fig. 7(a). (b) Tree placement with limited customization. (c) Tree placement with full customization. See Figs. 6 and 7(b). The width of the macro is 133 wiring tracks, and the number of successfully routed wires for each case is 10, 17, and 20, respectively.

(b), 30 matches were obtained, and the wirability improvement over transistor placement was 15 percent.

The macro designed with tree placement and full customization, case (c), is shown in **Figure 6.** Here the connectivity has been changed, and the maximum number of variable level matches, 74, was achieved. The macro wire length and via count were reduced in this case by 25 percent.

For the tree placement used in these cases, the right-hand portion of the second and third rails contains trees with

relatively few transistors, and thus offers a good view of the inter-tree wiring. This region is shown enlarged for cases (a) and (c) above, in **Figures 7(a)** and **7(b),** respectively. The increase in straight-line variable busing among trees and the reduction in number of vias obtained by tree customization is evident.

The vertical porosities for designs (a)–(c) above are shown in **Figure 8.** The porosities obtained were (a) 8 percent, (b) 13 percent, and (c) 15 percent, demonstrating the value of even limited tree customization. In each case, the horizontal porosity was 3 percent or less. The increased vertical porosity produced by customization, shown in Fig. 8, is a consequence of the increase in busing evident in Fig. 7(b).

Summary

ACORN is described; it is a system for the physical design of CVS macros which uses tree placement and tree customization. Decreased wire length and via count and increased porosity were demonstrated in the design of a 43-tree DCVS macro.

Acknowledgments

The authors wish to thank R. Brayton, J. Davis, W. Griffin, R. Kilmoyer, and S. White for helpful discussions and assistance.

References

1. L. G. Heller, W. R. Griffin, J. W. Davis and N. G. Thoma, "Cascode Voltage Switch Logic—A High Performance CMOS Logic Family," *Proceedings of the 31st International Solid State Circuit Conference*, San Francisco, 1984, pp. 16–17.
2. K. H. Khokhani and A. M. Patel, "The Chip Layout Problem—A Placement Procedure for LSI," *Proceedings of the 14th Design Automation Conference,* New Orleans, LA, 1977, pp. 291–297.
3. M. Hanan and J. M. Kurtzberg, "Placement Techniques," *Design Automation of Digital Systems,* M. A. Breuer, Ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1972, pp. 213–282.
4. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science* **220,** 671–680 (1983).
5. M. Burstein, S. J. Hong, and R. Pelavin, "Hierarchical VLSI Layout: Simultaneous Placement and Routing of VLSI Gate-Arrays," *Proceedings* of VLSI-83, North-Holland Publishing Co., Amsterdam, 1983, pp. 45–60.
6. M. D. F. Schlag, E. J. Yoffa, P. S. Hauge, and C.-K. Wong, "A Method for Improving Cascode-Switch Macro Wirability," submitted to *IEEE Trans. Computer-Aided Design of Integrated Circuits & Systems.*

Received December 29, 1983; revised March 28, 1984

Peter S. Hauge *IBM Research Division, P.O. Box 218, Yorktown Heights, New York 10598.* Dr. Hauge is in the logic synthesis group at the Thomas J. Watson Research Center, where he joined IBM in 1968. His research has included studies of the interaction of microwave, infrared, and visible radiation with semiconducting materials, and films. He is co-inventor of two automated ellipsometers for real-time film thickness measurement and surface studies. He has been involved with design automation since 1980, and for the past year has investigated the relationship of logical and physical design in cascode circuits. He received the B.S., M.S., and Ph.D. degrees in electrical engineering in 1961, 1963, and 1967, respectively, from the University of Minnesota, Minneapolis. Dr. Hauge is a member of the Optical Society of America.

Ellen J. Yoffa *IBM Research Division, P.O. Box 218, Yorktown Heights, New York 10598.* Dr. Yoffa received a B.S. (1973) and Ph.D. (1978) in physics from the Massachusetts Institute of Technology, Cambridge, where her area of study was theoretical solid state physics. In 1978, she joined the IBM Thomas J. Watson Research Center for postdoctoral work in the Semiconductor Science and Technology Department, where she investigated ballistic conduction in semiconductor devices and the physics of the laser annealing process. Since 1980, she has been a member of the Research staff in the Computer Science Department. Her research has involved the development of tools for VLSI physical design automation. Most recently, she has been studying the relationship of logical and physical design in cascode circuits. Dr. Yoffa is a member of the American Physical Society, Phi Beta Kappa, and Sigma Xi.